AStarTwice at SemEval-2021 Task 5: Toxic Span Detection using RoBERTa-CRF, Domain Specific Pre-Training and Self-Training

Thakur Ashutosh Suman* **Abhinav Jain***
Indian Institute of Technology (Indian School of Mines) Dhanbad, India
{ashutoshsuman99,jain.abhinav02}@gmail.com

Abstract

This paper describes our contribution to SemEval-2021 Task 5: Toxic Spans Detection. Our solution is built upon RoBERTa language model and Conditional Random Fields (CRF). We pre-trained RoBERTa on Civil Comments dataset, enabling it to create better contextual representation for this task. We also employed the semi-supervised learning technique of self-training, which allowed us to extend our training dataset. In addition to these, we also identified some pre-processing steps that significantly improved our F1 score. Our proposed system achieved a rank of 41 with an F1 score of 66.16%.

1 Introduction

In recent years there has been an exponential increase in the use of social network platforms. With rising abusive language and hate on such platforms, it is more important than ever to maintain online conversations constructive and inclusive. This problem can be tackled by filtering toxic comments/posts. The massive volume of data generated at a fast pace makes manually filtering each comment complicated and time-consuming. This process can be automated by modelling it as a supervised classification problem. A similar task was proposed in SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval) (Zampieri et al., 2019). Most of the top-ranked teams in this task used transformer language models (Liu et al., 2019a; Zhu et al., 2019; Pelicon et al., 2019; Wu et al., 2019) or an ensemble of CNN and RNN (Mahata et al., 2019; Mitrović et al., 2019) to classify the sentences.

The problem with the above approach is that it doesn't give moderators much knowledge about the reason for a sentence's toxicity. Highlighting toxic spans can help human moderators who frequently deal with long comments and prefer attribution rather than just an unexplained toxicity score. SemEval 2021 Task 5: Toxic Span Detection (Pavlopoulos et al., 2021) gave a chance to propose NLP systems to solve this problem. The task is concerned with developing systems that can recognise spans that contribute to the text's toxicity.

This task had a few challenges. Since the samples were from an online commenting platform, they were grammatically incorrect and consisted of many out of vocabulary words. The noisy and ambiguous structure of comments significantly hampers the performance of general NLP models. The training dataset had a little less than 8000 samples. Thus, there was a need to select systems that can produce meaningful results, even with a limited number of training samples. Undoubtedly, the hardest part is to identify spans that can account for the toxicity of the sample. The span could be as small as a single token and as large as the sample itself. The linguistic variations in the usage of words and phrases make such attribution even more difficult.

We formulated the task as a sequence tagging problem and used RoBERTa (Liu et al., 2019b), a pre-trained Transformer-based (Vaswani et al., 2017) language model as our base model. We further pre-trained RoBERTa on the Civil Comments Dataset as a masked language model (Devlin et al., 2018) to create a domain-specific model. We employed a Conditional Random Field (CRF) layer (Lafferty et al., 2001) for predicting the most probabilistic sequence of labels for each input sequence. We also applied a few pre-processing steps, which lead to significant performance improvements. Lastly, we leveraged the semi-supervised learning technique of self-training (Yarowsky, 1995; Liao and Veeramachaneni, 2009; Jurkiewicz et al., 2020) by training our model on the manually annotated

*Equal Contribution. Author order determined by a coin flip

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 875–880
Bangkok, Thailand (online), August 5–6, 2021.

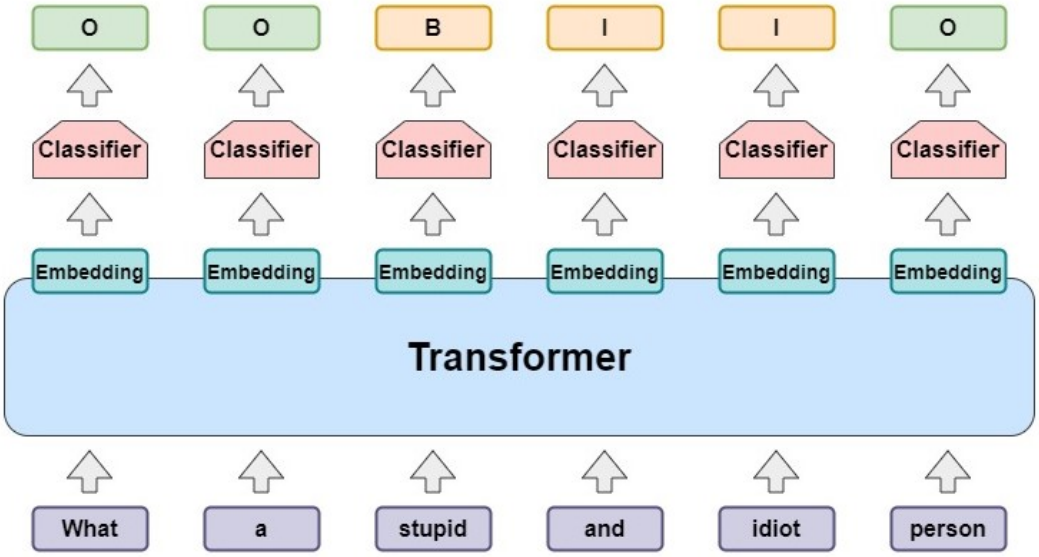

Figure 1: Our Model Architecture. We used RoBERTa as our transformer. Classifier constitutes two dense layers and a CRF layer with three labels.

dataset and using it to further extend the training set by generating toxic spans for other unannotated datasets. We have made our system's implementation available through GitHub[1].

The rest of the paper is organised as follows. Section 2 explains our model implementation in detail. Section 3 and 4 presents our experimental setup and achieved results, respectively. In section 4, we perform error analysis, followed by conclusions in the last section.

2 System Description

2.1 Pre-Training

Toxic comments have a different language construct from the general language. Their slang and obfuscated content (van Aken et al., 2018) make it difficult for the language models pre-trained on broader datasets to understand them. Similar to other domain-specific models (Beltagy et al., 2019; Lee et al., 2020; Paraschiv et al., 2020), we pre-trained the RoBERTa-base model on the Civil comments dataset using Masked Language Modelling (MLM) (Devlin et al., 2018) to provide the necessary domain knowledge and created our model RoBERTa(p). The original weights of RoBERTa-base served as the starting point for the pre-training. The pre-training was done for 0.2 million steps with a batch size of 32 and a learning rate of 2e-5.

[1] https://github.com/jain-abhinav02/Toxic_Spans_Detection

2.2 Fine-Tuning

We formulated the task as a token level sequence tagging problem where we classify each token as Begin, Inside or Outside (BIO scheme). Having begin and end tags helps formulate the notion of spans better and creates dependencies between various tokens of a toxic span (Singh et al., 2020), allowing it to perform better than other alternatives such as IO (Inside Outside).

Pre-Processing: We applied a few pre-processing steps before fine-tuning RoBERTa on the input text samples. First, we converted all the text samples to lowercase. We observed that punctuation marks did not add any significant information to the semantics of a sentence. Therefore, as a part of the data cleaning, punctuation marks such as commas and dashes were removed. We also collapsed multiple space characters into a single space.

Model: We provided the text samples as input to our pre-trained RoBERTa(p) model to get 768-dimensional contextual embeddings for each token. These contextual embeddings were passed through two dense layers of 512 and 128 dimensions, followed by a Conditional Random Fields (CRF) (Lafferty et al., 2001) layer with three labels (B-Begin, I-Inside or O-Outside). The CRF layer models the correlation between the labels predicted for the individual tokens. It receives the logits for each input token and predicts the most probabilistic sequence

Model	Tag	F1	Precision	Recall
RoBERTa	IO	0.6091	0.5831	0.7224
RoBERTa(p)	IO	0.6183	0.5841	**0.7408**
RoBERTa(p) + PP	IO	0.6376	0.6259	0.7264
RoBERTa(p) + PP + CRF	IO	0.6422	0.6323	0.7246
RoBERTa(p) + PP + CRF	BIO	0.6566	0.6512	0.7203
RoBERTa(p) + PP + CRF + ST(1)	BIO	0.6613	0.6537	0.7295
RoBERTa(p) + PP + CRF + ST(2)	BIO	**0.6634**	**0.6590**	0.7262

Table 1: Our model results on Test Set. RoBERTa(p) is our model pre-trained on domain-specific data. PP stands for Pre-processing. ST(1) and ST(2) represents self-training first and second iteration results, respectively.

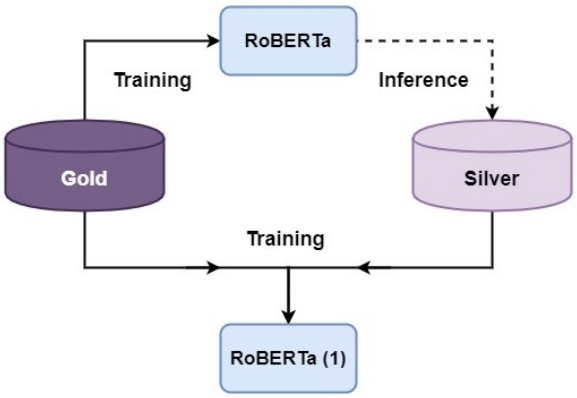

Figure 2: Self-Training of RoBERTa

of labels for each input sequence. Figure 1 shows our model architecture.

Post-Processing: The tokens decoded as B-Begin or I-Inside were marked as toxic. The character spans corresponding to these toxic tokens were added to the predicted spans. Two consecutive spans were merged if separated by at most five characters, provided all of them are non-alphabetic.

2.3 Self-Training

The best performing model on the manually annotated dataset (gold dataset) was used to generate toxic spans for the unannotated dataset. When selecting the unannotated data, we followed the process similar to the one used for creating the gold dataset (Pavlopoulos et al., 2021) that is, filter the most toxic samples (toxicity ≥ 0.80) from the Civil Comments dataset and select a random set of 10,000 samples. This process allowed the silver data to have similar toxicity distribution as the gold data. The newly generated annotations (silver dataset) were then used along with the gold dataset to train a new model. The model trained on the combined gold and silver dataset gave better performance (F1 score: 66.13%) than the one trained only on the gold dataset (F1 score: 65.66%). We repeated this process for one more iteration with another random set of 10,000 samples (F1 score: 66.34%). Figure 2 gives a simplistic idea of self-training.

3 Experimental Setup

Data: Each training example consisted of a text sample in English, and its ground truth toxic span provided as a list of character offsets (possibly empty). The posts were sampled from the publicly available Civil Comments dataset. The training set consisted of 7939 samples. We randomly sampled 20% of it as the development set. The test set for the evaluation phase had 2000 samples. In the training dataset, sample length varies from 1 to 421 tokens, with an average length of 47 tokens when tokenized using the RoBERTa-base tokenizer. Nearly 10% of all tokens in the training dataset are marked as toxic. The mean span length is 17.5 characters and 485 samples have empty spans. Further details about the dataset can be found in the task description paper (Pavlopoulos et al., 2021).

Evaluation measures: The evaluation measure for a sample is the F1 score between the predicted spans and the ground truth spans as defined in the SemEval-2021 Task 5 paper (Pavlopoulos et al., 2021). The overall score is obtained by taking the mean of the F1 score of all samples in the test set.

Implementation details: Since the text samples were lengthy, we split them into sentences. We used spacy's sentence tokenizer for this purpose. Adam (Kingma and Ba, 2014) was used as the optimizer with a learning rate of 1e-5. The batch size was set to 32. All the model variants were trained for at most five epochs. We used the RoBERTa-base tokenizer and model implementation provided

Original Spans	Predicted Spans
Small Sentences	
[Sick pervert]	Sick pervert
[bloody wankers]	bloody wankers
Rare Words	
Just another day in the Trump **[kleptocracy]**.	Just another day in the Trump kleptocracy.
Should ha taken his weapon n **[pulverized]** him	Should ha taken his weapon n pulverized him
Non-swear words	
When you're ready to place a monetary bet on whether these guys get charged with terrorism, lemme know. Otherwise, I know you're **[all bluster and no balls]**.	When you're ready to place a monetary bet on whether these guys get charged with terrorism, lemme know. Otherwise, I know you're all bluster and no **[balls]**.
Yes, I agree. We need an electoral college for Washington so that Seattle no longer votes for Washington. My goodness,they are a **[bunch of loons]**!	Yes, I agree. We need an electoral college for Washington so that Seattle no longer votes for Washington. My goodness,they are a bunch of **[loons]**!

Table 2: Common errors made by our model. Spans are made bold and are present in between square brackets.

by Huggingface[2]. The RoBERTa model was followed by two dense layers with 512 and 128 units with relu (Agarap, 2018) as the activation function and a dropout rate of 0.1. The output layer had two or three labels depending on the tagging scheme. We applied the post-processing steps mentioned in section 2.2 for all the model variants.

4 Results

Table 1 shows that our RoBERTa(p) model outperforms the original RoBERTa model. As suggested earlier, domain-specific pre-training allows the model to understand the language construct of toxic comments better. Additionally, we observe a significant increase in performance by adding preprocessing steps as it makes the model more robust to the noise present in the text samples. Adding the CRF layer further improves the F1 score by eliminating the problem of independent label prediction. It is evident from table 1 that the BIO tagging scheme performs better than the IO tagging scheme when working with CRF, suggesting it can better understand the span nature of the output. Finally, using two rounds of self-training helped us achieve our best F1 score, 66.34%[3].

One interesting observation that can be drawn from Table 1 is that for almost all the models, the recall remains constant and improvement in F1 is due to improvement in precision. The constancy of recall indicates that few spans are not captured as toxic by any of the models.

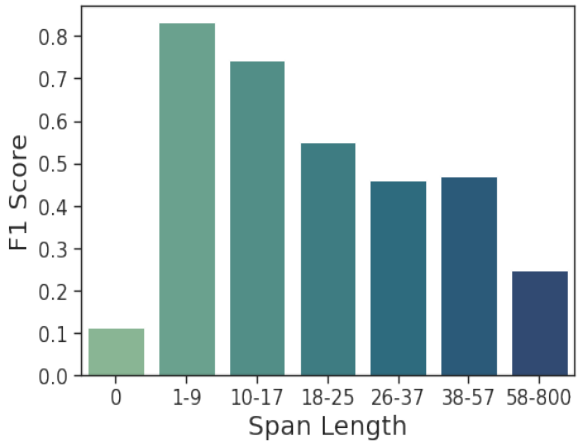

Figure 3: Distribution of F1 Score across different span lengths. Here span length refers to the total length of the toxic span in each sample. The value represented is the mean F1 score of all the text samples whose toxic span length falls in a particular range.

5 Error Analysis

Figure 3 shows the variation of the F1 score across different toxic span lengths on the test dataset. Our model achieved a very high F1 score when one (Span Length 1-9, Mean F1 Score: 83.17%) or two (Span Length 10-17, Mean F1 Score: 74.44%) words are marked as toxic in a text sample. As the number of characters marked as toxic increases,

[2] https://github.com/huggingface/transformers

[3] We achieved an F1 score of 66.16% in the official competition. However, our model achieved a even higher F1 score 66.34%, when the predictions of a different epoch were used for evaluation.

the F1 score falls drastically, reaching as low as 24.82% when more than 58 characters are marked as toxic. There are two main reasons for this. First, it is easier for the model to capture short-term dependencies than long-term dependencies. Second, only 10% of the training data has a span length of more than 25 characters making the model less equipped to capture such toxic spans.

To investigate our model's most problematic cases, we analysed the samples for which our model gave a zero F1 score. There were 447 such samples, of which 349 samples did not have any toxic span in the ground truth. This is also reflected in Figure 3, as the mean F1 score of all the samples with zero span length is 11.42%. Further analysis revealed that our model tends to mark those tokens as toxic, which were frequently found to be toxic elsewhere. A few samples with empty toxic spans had doubtful gold annotations. However, in other samples, our model failed to capture the sentence's context precisely and predicts tokens that were not used in a toxic sense.

Table 2 shows other standard errors our model makes. It seems that our model has a problem with small sentences. More often than not, it misses the toxic span present in it and returns an empty span. A similar case occurs when it encounters text samples with rare toxic words. These words may be present in very few examples or be completely absent from the training dataset, making our model less endowed to understand them. Other than these, our model sometimes misses the non-swear words in a toxic span.

6 Conclusion

This paper described our system developed for SemEval-2021 Task 5: Toxic Span Detection. We built our solution on the RoBERTa language model and Conditional Random Fields (CRF). Though RoBERTa alone can achieve great results, we highlighted the benefits of using external datasets and the performance improvements it can help us achieve. We pre-trained RoBERTa on the Civil Comments dataset to impart domain-specific knowledge to it. We also employed the semi-supervised learning technique of self-training to extend our training dataset. In addition to these, we also discovered some pre-processing steps that significantly improved our F1 score. Experimenting with different tagging schemes, we found out that the BIO scheme works the best with CRF.

In future, we plan to experiment with other language models such as T5 (Raffel et al., 2019), XL-Net (Yang et al., 2019) and DeBERTa (He et al., 2020). The system could also benefit from the addition of syntactic and semantic features at the word and sentence level.

References

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. Applicaai at semeval-2020 task 11: On roberta-crf, span cls and whether self-training helps them. *arXiv preprint arXiv:2005.07934*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Wenhui Liao and Sriharsha Veeramachaneni. 2009. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, pages 58–65, Boulder, Colorado. Association for Computational Linguistics.

Ping Liu, Wen Li, and Liang Zou. 2019a. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Debanjan Mahata, Haimin Zhang, Karan Uppal, Yaman Kumar, Rajiv Ratn Shah, Simra Shahid, Laiba Mehnaz, and Sarthak Anand. 2019. MIDAS at SemEval-2019 task 6: Identifying offensive posts and targeted offense from Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 683–690, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Jelena Mitrović, Bastian Birkeneder, and Michael Granitzer. 2019. nlpUP at SemEval-2019 task 6: A deep neural language model for offensive language detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 722–726, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Andrei Paraschiv, Dumitru-Clementin Cercel, and Mihai Dascalu. 2020. Upb at semeval-2020 task 11: Propaganda detection with domain-specific trained bert. *arXiv preprint arXiv:2009.05289*.

John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. 2021. Semeval-2021 task 5: Toxic spans detection (to appear). In *Proceedings of the 15th International Workshop on Semantic Evaluation*.

Andraž Pelicon, Matej Martinc, and Petra Kralj Novak. 2019. Embeddia at SemEval-2019 task 6: Detecting hate with neural network and transfer learning approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 604–610, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Paramansh Singh, Siraj Sandhu, Subham Kumar, and Ashutosh Modi. 2020. newssweeper at semeval-2020 task 11: Context-aware rich feature representations for propaganda classification. *arXiv preprint arXiv:2007.10827*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Zhenghao Wu, Hao Zheng, Jianming Wang, Weifeng Su, and Jefferson Fong. 2019. BNU-HKBU UIC NLP team 2 at SemEval-2019 task 6: Detecting offensive language using BERT model. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 551–555, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Jian Zhu, Zuoyu Tian, and Sandra Kübler. 2019. Umiu@ ling at semeval-2019 task 6: Identifying offensive tweets using bert and svms. *arXiv preprint arXiv:1904.03450*.